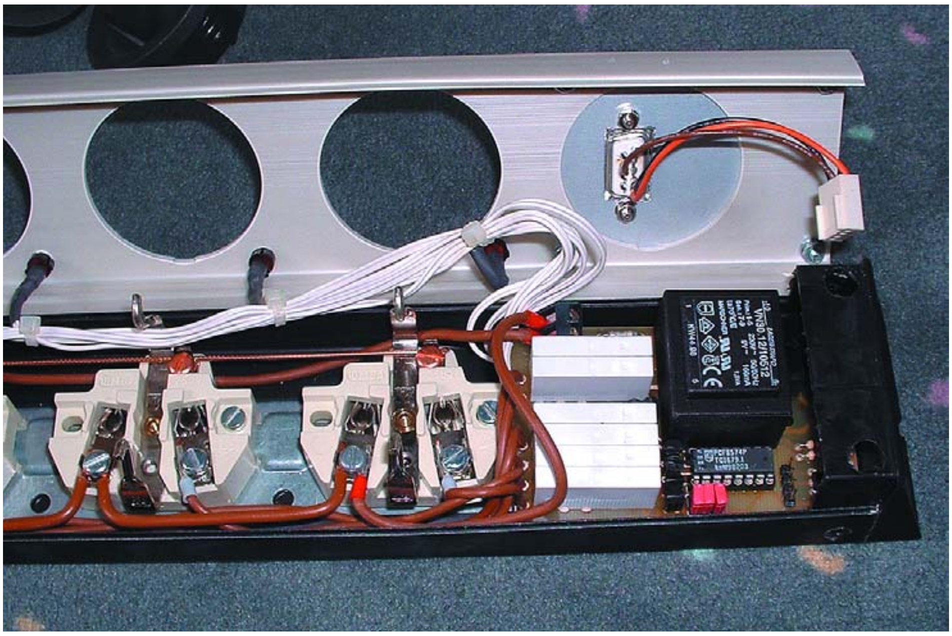
Switched-Outlet Power Bar with RS232 Control

Remote switching using RS232 and I^{2}C

Design by W. Frank

This circuit can be built into a power bar and used to remotely switch six socket-outlets. It is controlled via an RS232 interface and uses an I^{2}C-bus IC (type PCF8574P) to decode the RS232 signals.



A remotely switchable power bar controlled by a PC via the RS232 serial interface can be realized with a minimum of hardware and software drivers (for use under Windows, DOS and Linux) by using an IC to decode the RS232 signals and individually control each of the socket-outlets.

The 8-bit I/O expander

As can be seen from the block diagram in **Figure 1**, the IC in question, a type PCF8574, is basically a serial-to-parallel converter. What it essentially does is to take serial data from the I^{2}C bus and feed them into a shift register with an 8-bit output latch. Of course, this is nothing special, and it can be done using standard components.

However, the PCF8574 has a number of useful features that make it a good choice this application. For instance, it is addressable. Three bits of the PCF8574's address are freely selectable, while the four most significant bits are either '0100' (for the PCF8574) or '0111' (for the PCF8574A version). This means that in principle, up to 16 such devices can be independently operated using a single I^{2}C bus.

The PCF8574 also has an internal power-on reset circuit, so the output latch assumes a defined state after the circuit is switched on. The I^{2}C bus signals pass through an input filter and thus arrive at the control logic of the IC free of interference.

Ports P0-P7 are not just outputs. They operate in a 'quasi-bidirectional' mode, so the levels on P0-P7 can not only be set in the write mode, but also read in the read mode, without having to use a control signal to select the data direction. However, this capability is not used in this circuit.

The rest of the hardware

Besides the PCF8574, there's nothing particularly remarkable about the schematic diagram of the circuit shown in **Figure 2**. The base address of the IC can be set using three jumpers (JP1-JP3). If you are sure you will never want to change the address, you can omit the pull-up resistors (R1-R3) and simply connect the IC pins directly to ground or the positive supply voltage.

The I^{2}C bus is emulated by the RS232 interface of the PC, which is connected to K6. Besides the ground lead, only the DTR and RTS leads are used for this purpose. The diodes provide level conversion from ±12 V (RS232) to the +5 V and 0 V that the PCF8574 needs for the clock and data signals. This simple solution permits only one-way data traffic, from the PC to the PCF8574, but that's all we need here.

Six of the chip's eight I/O pins are used here, with each one controlling a relay via a driver transistor. Freewheeling diodes are connected in parallel with the coils of the relays. The supply voltage for the relays is approximately 12 V, which is taken directly from the 'raw' DC voltage obtained by rectifying the secondary voltage of power supply transformer Tr1 using B1. A voltage regulator provides +5 V for the PCF8574. LEDs with 1.2-kΩ series resistors can optionally be connected from the junctions of the relay coils and the collectors of the driver transistors to the +12-V line, as shown on the schematic diagram, in order to indicate the status of each of the relays.

Construction and fitting

The layout of the printed circuit board for this project is shown in **Figure 3**. The dimensions of the circuit board have been chosen to allow it to be easily fitted into commonly available multi-way power bars. Naturally, it must be possible to take apart the power bar. Cheap power bars, which are usually hot-welded together, are thus not suitable. Also, the individual socket-out-

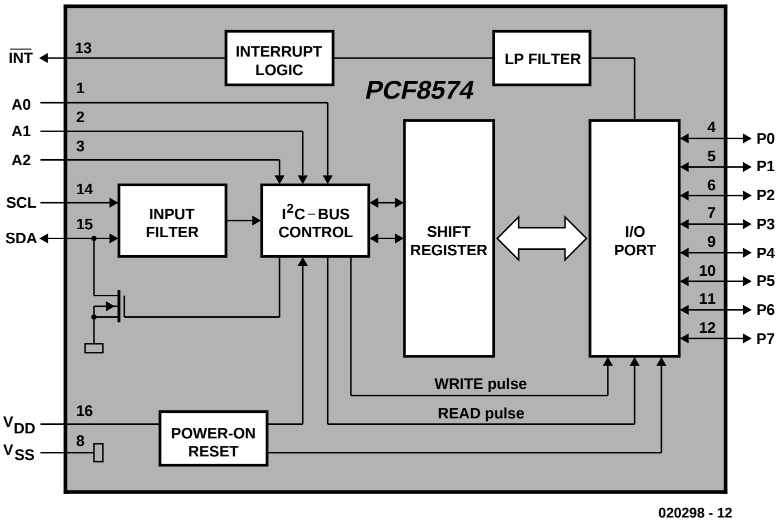


Figure 1. Functional block diagram of the PCF8574 (Philips Semiconductors).

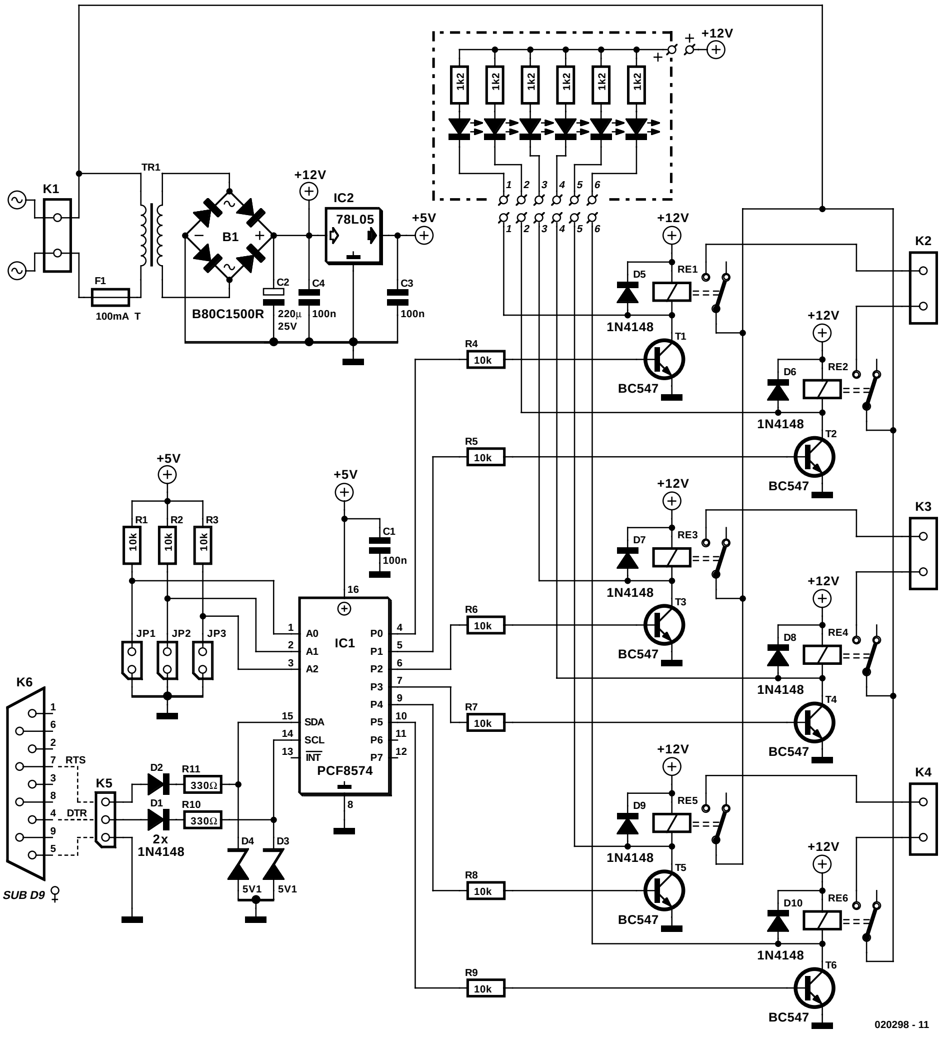


Figure 2. The circuit combines the I^{2}C chip with a serial interface and six output relays.

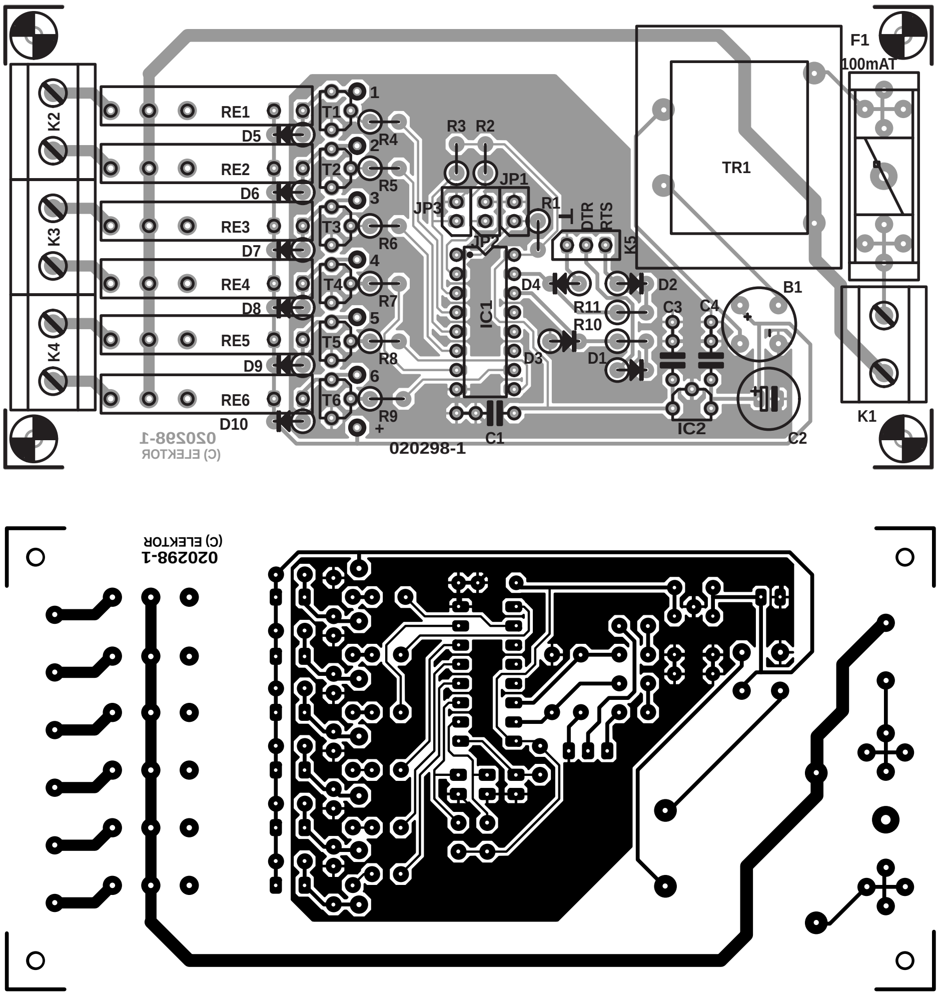


Figure 3. The printed circuit board fits exactly into a power bar enclosure.

COMPONENTS LIST

Resistors:

R1-R9 = 10kΩ

R10,R11 = 330Ω

Capacitors:

C1,C3,C4 = 100nF

C2 = 220µF 25V radial

Semiconductors:

B1 = B80C1500 in round case (80V piv, 1.5A continuous)

D1,D2,D5-D10 = 1N4148

D3,D4 = zener diode 5.1V, 500mW

T1-T6 = BC547

IC1 = PCF8574 or PCF8574A

IC2 = 78L05

Miscellaneous:

JP1,JP2,JP3 = jumper or wire link

K1-K4 = 2-way PCB terminal block, lead pitch 7.5mm

K5 = 3-way SIL pinheader

K6 = 9-way sub-D socket (female), chassis mount

TR1 = mains transformer, 9V 1.5VA, (e.g., Hahn BV EI 302 2021)

RE1-RE6 = relay, Finder type 34.51.7.012.00 (12V) (Conrad Electronics # 504459)

F1 = fuse, 100mA, slow, with fuse holder and protective cover

Optional: six 1.2kΩ resistors and six LEDs

PCB, order code **020298-1** (see Readers Services page)

Disk, order code **020298-11,** or Free Download)

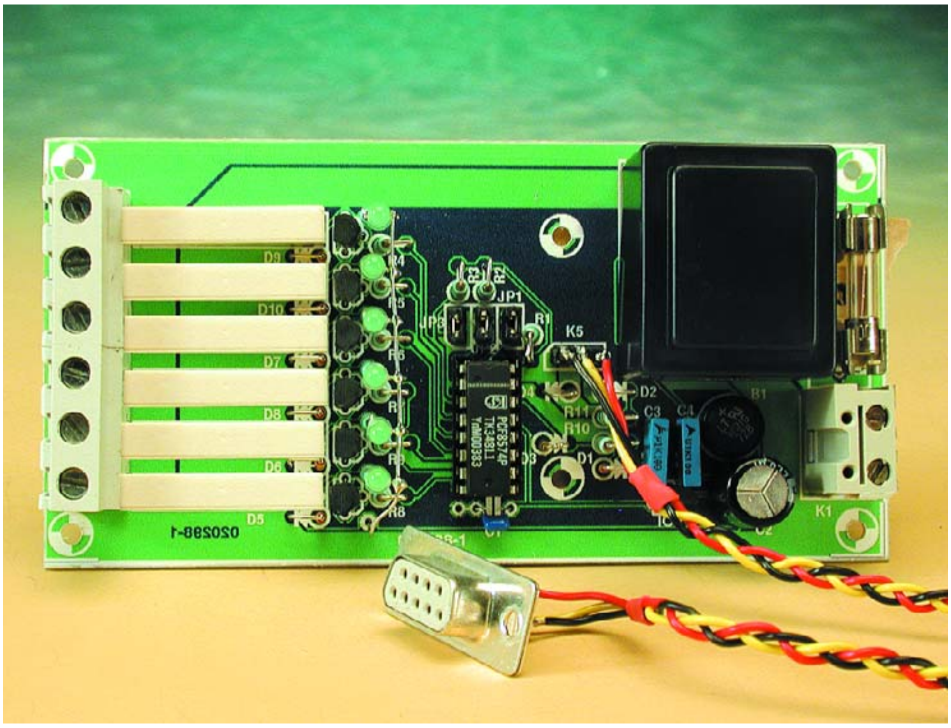
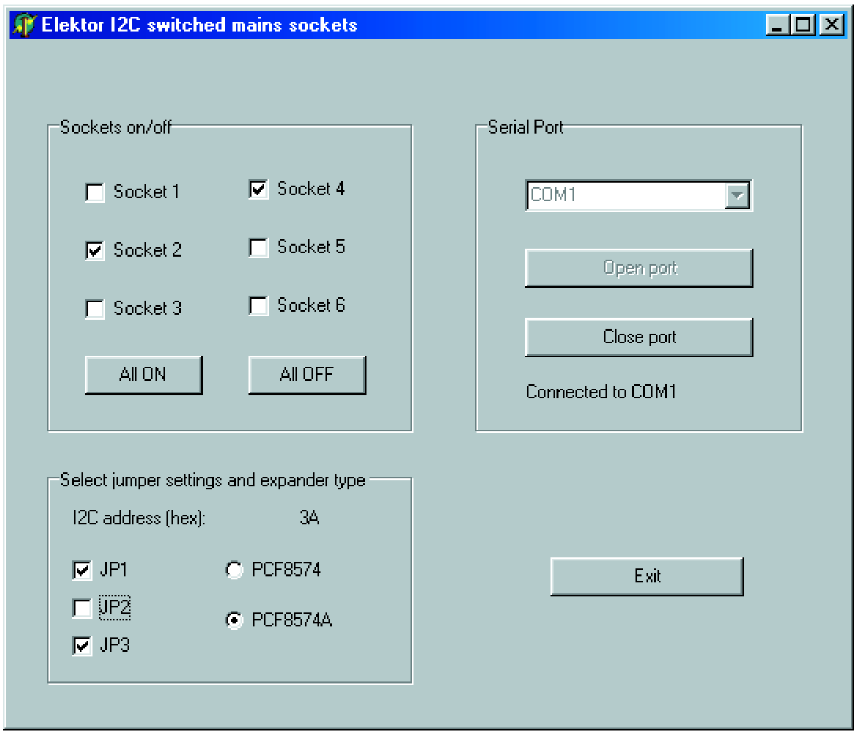


Figure 4. The fully assembled *Elektor Electronics* prototype circuit.

lets must be interconnected using flexible wiring, rather than solid strips of metal. To make room for the control circuit, remove the first of the socket-outlets. The circuit board has four drilled mounting holes to allow it to be securely attached to the power bar enclosure.

The photo at the head of the article shows the author’s prototype. The circuit board in this photo is not the same as the *Elektor Electronics* circuit board (see **Figure 4**).

Assembling the circuit board is perfectly straightforward. All of the passive components and discrete semiconductors are fitted vertically. As already noted, the three jumpers can be replaced by wire links to ground or the 5-V supply voltage. In



Sockets on/off
☐ Socket 1
☑ Socket 2
☐ Socket 3
☑ Socket 4
☐ Socket 5
☐ Socket 6

Serial Port
COM1
Connected to COM1

Select jumper settings and expander type
I2C address (hex): 3A
☑ JP1
☐ JP2
☑ JP3
○ PCF8574
● PCF8574A

Figure 5. Using a Delphi 5 window to control the power bar under Windows.

this case, R1-R3 can be omitted. The narrow mains relays made by the Swiss company Finder allow the board to be kept very compact.

Drivers

The project includes source code and .exe files for Windows (a Delphi application), Linux and DOS (both written in C). All of these programs can be downloaded free of charge from the *Elektor Electronics* website, including the source code. They can also be obtained from Readers Services on diskette (order number **020298-11**).

The core of the program consists of an emulation of the I^{2}C bus using two leads of the 'normal' serial PC interface. The DTR lead drives the serial I^{2}C clock line SCL, while the RTS lead drives the I^{2}C data line SDA. Since the PC cannot receive any data from the I^{2}C chip, the software simply ignores Acknowledge signals from the PCF8574.

After the PC software is started, or after a data packet has been sent, the bus is in the idle state, with both of the I^{2}C lines high. The bus master initiates a data transfer by generating a negative edge on SDA while SCL is high (Start condition). After this, SCL also goes low. The seven address bits, followed by the R/W bit, are transferred sequentially by a series of rising edges on the SCL line. The level on the SDA line is only allowed to change while the SCL line is low. In this application, the R/W bit is naturally always 0 (= write).

Normally, the slave is allowed to confirm the receipt of a valid address by sending Acknowledge. Although this is not the case here, the master must still generate an extra clock pulse, which the slave interprets as confirmation of the Acknowledge.

After this, the slave is ready to receive the eight data bits. Each bit is directly associated with one of the IC outputs. The data bits are followed by an extra SCL pulse to confirm the Acknowledge the slave thinks it has sent, after which the master terminates the transfer with a positive edge on SDA while SCL is high.

Control

The functions of `SERIALDLL.DLL`, which has been extensively described in the March 2003 issue of *Elektor Electronics*, are used to drive the serial interface. This DLL, which also forms part of the software for this project that is available by download or on diskette, should be copied to the directory \WINDOWS\SYSTEM.

The actual program, `SOCKETS`, uses the functions SetDTR, Reset-DTR, SetRTS and ResetRTS. In addition, COMPortExist determines which serial ports are available, and OpenCOM & CloseCOM are necessary to active and deactivate the port to which the power bar is connected.

The most important procedure is SendI2C, which sends a data byte to a particular I^{2}C address. A glance at the source codes shows that all that this involves is driving the two lines in the proper sequence. A brief delay is introduced after each level change on the lines in order to keep the data transmission rate on the bus within allowable limits.

Figure 5 shows the window displayed by the sample Delphi 5 program, which is available as source code. It is fairly self-explanatory. At the beginning, the desired COM port is selected and opened, after which the address of the I^{2}C chip is selected. This depends on the IC type (the PCF8574 has a base address of $20, while the -A version has a base of address of $38) and the configuration of jumpers JP1-JP3. Once these settings have been made, the individual socket-outlets can be switched on or off as desired. To stop the program, close the serial port or simply click on Exit.

(020298-1)

Free Downloads

- source code files and .exe files for Windows (Delphi application), Linux and DOS (both written in C).
  File number: **020298-11.zip**
- circuit board layout in PDF format.
  File number: **020298-1.zip**

www.elektor-electronics.co.uk/dl/dl.htm
(select month of publication)